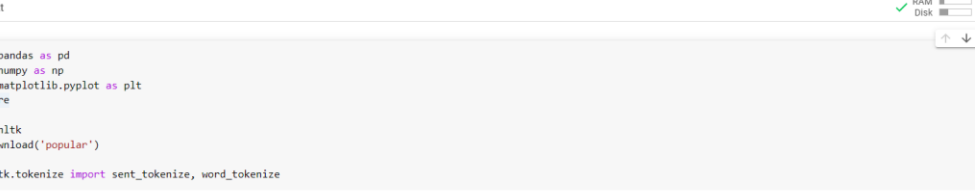


Mehmet Acikgoz

ICP-03 TWITTER SENTIMENT ANALYSIS

Importing the libraries



The screenshot shows a Jupyter Notebook environment with the following code in the first cell:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re

import nltk
nltk.download('popular')

from nltk.tokenize import sent_tokenize, word_tokenize
```

The output of the cell shows the progress of downloading the NLTK data collection and various packages, all of which are already up-to-date:

```
[nltk_data] Downloading collection 'popular'
[nltk_data]
[nltk_data] | Downloading package cmudict to /root/nltk_data...
[nltk_data] |   Package cmudict is already up-to-date!
[nltk_data] | Downloading package gazetteers to /root/nltk_data...
[nltk_data] |   Package gazetteers is already up-to-date!
[nltk_data] | Downloading package genesis to /root/nltk_data...
[nltk_data] |   Package genesis is already up-to-date!
[nltk_data] | Downloading package gutenberg to /root/nltk_data...
[nltk_data] |   Package gutenberg is already up-to-date!
[nltk_data] | Downloading package inaugural to /root/nltk_data...
[nltk_data] |   Package inaugural is already up-to-date!
[nltk_data] | Downloading package movie_reviews to
[nltk_data] |   /root/nltk_data...
[nltk_data] |   Package movie_reviews is already up-to-date!
[nltk_data] | Downloading package names to /root/nltk_data...
[nltk_data] |   Package names is already up-to-date!
[nltk_data] | Downloading package shakespeare to /root/nltk_data...
[nltk_data] |   Package shakespeare is already up-to-date!
[nltk_data] | Downloading package stopwords to /root/nltk_data...
[nltk_data] |   Package stopwords is already up-to-date!
[nltk_data] | Downloading package treebank to /root/nltk_data...
[nltk_data] |   Package treebank is already up-to-date!
[nltk_data] | Downloading package twitter_samples to
```

Reading the data

[illegible]

Data Sampling for Reducing Computation Burden

ICPO3_SentimentAnalysisWithNLTK.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Editing

To get some percentage of the data as sample for modelling in terms of reducing the computation burden

```
[4] sample_size = int(np.array(df.shape[0])*SAMPLE_PERCENTAGE)
df = df.sample(sample_size, replace=False)
print(f"Sample size is {sample_size}")
print(f"Sample data dimension is {df.shape}")
```

Sample size is 6392
Sample data dimension is (6392, 2)

df.head(FIRST_N_RECORDS)

	label	tweet
10753	0	@user oooo tomorrow
7895	0	i opened "my" #rollerblade season, today! #summer #rollerblading #belgrade #adaciganija
30918	0	father day
15004	0	#type 40 min to go #super #oneplus3 @user @user @user
20044	0	while i am eating breakfast, i hear the sad sounds of packing. this vacation has seemed shorter than the others.
29752	0	@user oh noooo, don't re-write! you'll be fine just going with the flow, it will be epic. :) #oustudents16 #cantsleeptnight
14281	0	happy 6th bithday junior i hope you have the best day! @user bithday #cristiano #c77 happy bithday to mel... bithday day holidays #monday #menmodels
477	0	happy bithday to mel... bithday day holidays #monday #menmodels
6277	0	@user i freaking #love #this! #selfies #alldayeveryday if it makes you i #goodhairday? #makeup #onpoint? selfie
31401	0	@user : find out which nationalities are happy for most of their days -

Dropping the duplicate records:

ICPO3_SentimentAnalysisWithNLTK.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Editing

```
[6] # Removing duplicate records
def drop_duplicate_records(df):
    num_of_duplicate_records = len(df)-len(df.drop_duplicates(inplace=False))
    df.drop_duplicates(inplace=True)
    print(f"{num_of_duplicate_records} duplicate records are dropped.\n\nThe number of unique records is {df.shape[0]}")
```

drop_duplicate_records(df)

280 duplicate records are dropped.
The number of unique records is 6112

```
[8] # To explore the data
df.describe()
```

	label	tweet
count	6112	6112
unique	2	6112
top	0	be with what you have, while still #working for what you want #mavikbeb
freq	5719	1

```
[9] # To check the missing/null values
df.isnull().sum()
```

	label	tweet
label	0	0
tweet	0	0
dtype:	int64	

Data Cleaning:

ICPO3_SentimentAnalysisWithNLTK.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Editing

Removing the Twitter Handles and Hashtags

```
[10] def remove_pattern(input_txt, pattern):
    ...
    This removes the twitter handles from the pandas dataframe
    https://stackoverflow.com/questions/50830214/remove-usernames-from-twitter-data-using-python
    ...
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)
    return input_txt

# To remove the twitter handles
df['tweet'] = np.vectorize(remove_pattern)(df['tweet'], "@[\w]*")
# df['tweet'] = np.vectorize(remove_pattern)(df['tweet'], "#[\w]*")
df.head(FIRST_N_RECORDS)
```

	label	tweet
10753	0	oooo tomorrow
7895	0	i opened "my" #rollerblade season, today! #summer #rollerblading #belgrade #adaciganlija
30918	0	father day
15004	0	#hype 40 min to go #super #oneplus3
20044	0	while i am eating breakfast, i hear the sad sounds of packing. this vacation has seemed shoor than the others.
29752	0	oh noooo, don't re-write! you'll be fine just going with the flow, it will be epic. :) #oustudents16 #cantsleeptonight
14281	0	happy 6th bihday junior i hope you have the best day! bihday #cristiano #cr7

Tokenization

ICPO3_SentimentAnalysisWithNLTK.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Editing

```
[12] text = "".join(df.tweet.to_list())
print(len(text))
print(text)
```

516947

oooo tomorrow i opened "my" #rollerblade season, today! #summer #rollerblading #belgrade #adaciganlija father day #hype 40 min to go #super #oneplus3 while

Sentence Tokenization

```
# Sentence Tokenization
sent_tokens = sent_tokenize(text)

print(f"Number of sentences: {len(sent_tokens)}")
for sentence in sent_tokens[:FIRST_N_RECORDS]:
    print(sentence)
```

Number of sentences: 4470

oooo tomorrow i opened "my" #rollerblade season, today!
#summer #rollerblading #belgrade #adaciganlija father day #hype 40 min to go #super #oneplus3 while i am eating breakfast, i hear the sad sounds of packing.
this vacation has seemed shoor than the others.
oh noooo, don't re-write!
you'll be fine just going with the flow, it will be epic.
;) #oustudents16 #cantsleeptonighthappy 6th bihday junior i hope you have the best day! bihday #cristiano #cr7
brithday day holidays #monday #menmodels i freaking #love #this!
#selfies #alldayeveryday if it makes you !
#makeup #onpoint?

```
ICPO3_SentimentAnalysisWithNLTK.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
# Word tokenization
word_tokens = word_tokenize(text)
print(f"Number of sentences: {len(word_tokens)}")
for word in word_tokens[:FIRST_N_RECORDS]:
    print(word)

Number of sentences: 101970
0000
tomorrow
A^A^A^A^A^A^A^A^A^A
i
opened
..
my
..
#
rollerblade

Cleaning the data (#removing punctuation, numbers, special characters)

[15] from nltk.corpus import stopwords
      from string import punctuation

      stop_words = set(stopwords.words('english')+list(punctuation))
      len(stop_words)
      print(stop_words)

['$he', 'than', 'it's', 'ma', 'hadn't', 'some', 'all', ':', 'needn't', '/', 'himself', 'where', 'out', 'how', 'aren', '%', 'your', ')', 'him', 'her', 'themselves', "didn't", 'doing', 'only', '
[16] cleaned_tweet_list = []
```

Rearranging the data after cleaning from the stopwords, punctuation and special characters

The screenshot shows a Jupyter Notebook environment with the following components:

- Top Bar:** Includes the Jupyter logo, the file name "ICPO3_SentimentAnalysisWithNLTK.ipynb", and standard menu items (File, Edit, View, Insert, Runtime, Tools, Help). On the right, there are icons for Comment, Share, Settings, and a user profile.
- Code Editor:** Contains Python code for cleaning tweets. The code defines a function to clean a list of tweets by removing non-alphabetic characters and stop words. It then applies this function to a DataFrame column and prints the results.
- Output:** The code execution results in a DataFrame with two columns: "label" and "tweet". The "label" column contains binary values (0 or 1), and the "tweet" column contains the cleaned text of the tweets.

Code Snippet:

```
cleaned_tweet_list = []
for sentence in df.tweet.to_list():
    words = [word for word in word_tokenize(sentence.lower()) if ((word.isalpha() == True) & (word not in stop_words))]
    newsentence = (" ".join(words)).strip()
    cleaned_tweet_list.append(newsentence)

for sentence in cleaned_tweet_list[:FIRST_N_RECORDS]:
    print(sentence)
```

DataFrame Output:

	label	tweet
0	0	oooo tomorrow
1	0	opened rollerblade season today summer rollerblading belgrade adaciganlija
2	0	father day
3	0	hype min go super
4	0	eating breakfast hear sad sounds packing vacation seemed shoer others

Dropping duplicate records:

ICPO3_SentimentAnalysisWithNLTK.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Editing

```
[18] drop_duplicate_records(df)
```

140 duplicate records are dropped.
The number of unique records is 5972

```
df.describe()
```

	label	tweet
count	5972	5972
unique	2	5971
top	0	
freq	5594	2

```
[20] df.label.value_counts()
```

```
0    5594  
1     378  
Name: label, dtype: int64
```

```
[21] import seaborn as sns  
fig= plt.figure(figsize=(3,5))  
sns.countplot(x='label', data = df);
```

/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
import pandas.util.testing as tm

Target Value Distribution:

ICPO3_SentimentAnalysisWithNLTK.ipynb

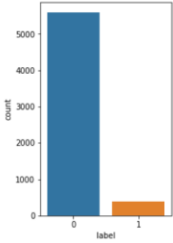
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Editing

```
import seaborn as sns  
fig= plt.figure(figsize=(3,5))  
sns.countplot(x='label', data = df);
```

/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
import pandas.util.testing as tm



```
[22] # This is imbalanced data. There are some ways to deal with this problem.
```

```
[23] words_cleaned = word_tokenize("".join(cleaned_tweet_list))  
print(f"The number of words is {len(words_cleaned)}\n")  
for word in words_cleaned[:FIRST_N_RECORDS]:  
    print(word)
```

Data Preprocessing and Frequency Distribution

ICPO3_SentimentAnalysisWithNLTK.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

RAM Disk Editing

+ Code + Text

```
[23] words_cleaned = word_tokenize("".join(cleaned_tweet_list))
print(f"The number of words is {len(words_cleaned)}\n")
for word in words_cleaned[:FIRST_N_RECORDS]:
    print(word)
```

The number of words is 38665
oooo
tomorrowopened
rollerblade
season
today
summer
rollerblading
belgrade
adaciganlijafather
daytype

```
from nltk.probability import FreqDist
freq_dist= FreqDist(words_cleaned)
freq_dist.most_common(10)
```

```
[('day', 346),
 ('love', 336),
 ('amp', 326),
 ('like', 184),
 ('life', 164),
 ('new', 158),
 ('happy', 150),
 ('today', 135),
 ('see', 128),
 ('get', 126)]
```

```
[25] fig= plt.figure(figsize=(12,6))
```

ICPO3_SentimentAnalysisWithNLTK.ipynb

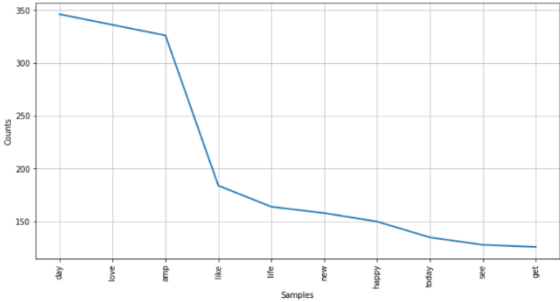
File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

RAM Disk Editing

+ Code + Text

```
fig= plt.figure(figsize=(12,6))
freq_dist.plot(10)
```

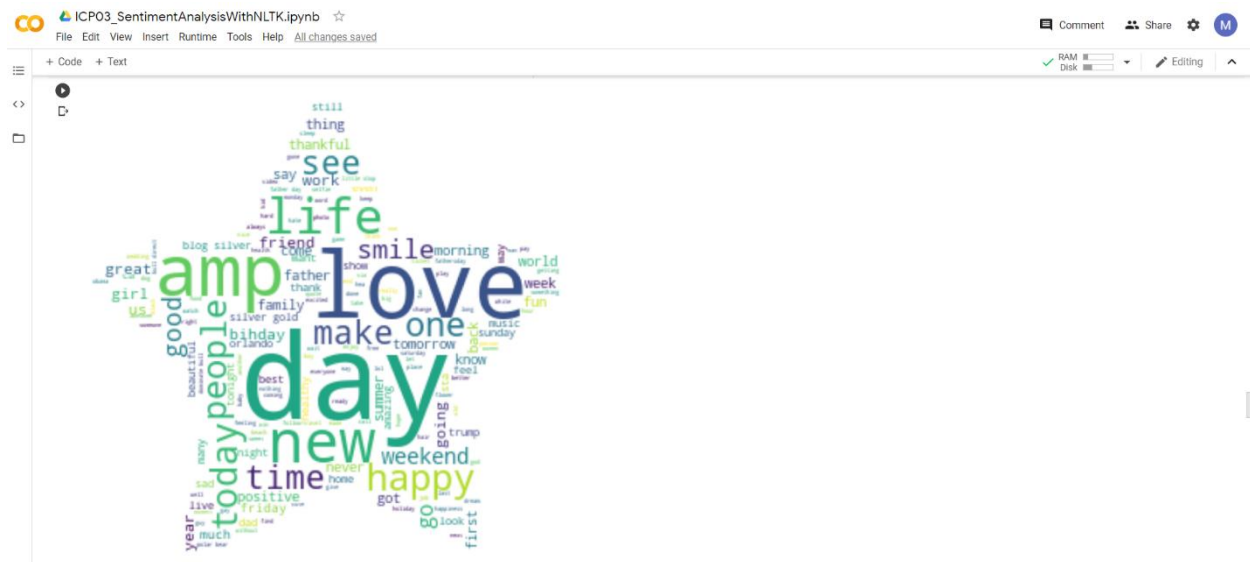
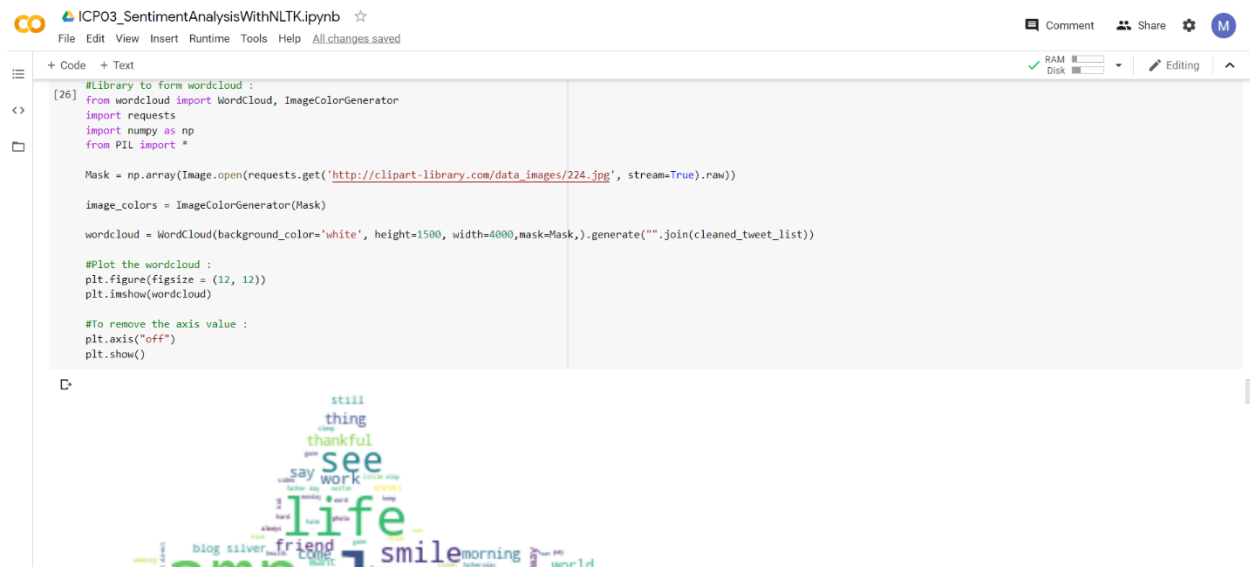


Word	Count
day	346
love	336
amp	326
like	184
life	164
new	158
happy	150
today	135
see	128
get	126

```
[26] #Library to form wordcloud :
from wordcloud import WordCloud, ImageColorGenerator
import requests
import numpy as np
from PIL import *
```

```
Mask = np.array(Image.open(requests.get('http://clipart-library.com/data/images/224.jpg', stream=True).raw))
```

WordCloud Generated from the data



Stemming, Lemmatization, Part-of-Speech:

```
ICPO3_SentimentAnalysisWithNLTK.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[27] # Stemming
from nltk.stem import PorterStemmer
ps = PorterStemmer()
[(word, ps.stem(word)) for word in words_cleaned[:FIRST_N_RECORDS]]

[28] # Lemmatization
from nltk import WordNetLemmatizer
wnl = WordNetLemmatizer()
[(word, wnl.lemmatize(word)) for word in words_cleaned[:FIRST_N_RECORDS]]

[29] # Part-of-Speech
```

```
ICPO3_SentimentAnalysisWithNLTK.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[29] # Part-of-Speech
tag = nltk.pos_tag(words_cleaned[:FIRST_N_RECORDS])
tag

[30] for sentence in df.tweet.to_list()[:FIRST_N_RECORDS]:
    print(nltk.pos_tag(word_tokenize(sentence)))

[31] X = df.tweet
    y = df.label

[32] X.head(FIRST_N_RECORDS)
```


ICPO3_SentimentAnalysisWithNLTK.ipynb

```

[30] for sentence in df.tweet.to_list()[0:FIRST_N_RECORDS]:
    print(nltk.pos_tag(word_tokenize(sentence)))

[('oooo', 'NN'), ('tomorrow', 'NN')]
[('opened', 'VBN'), ('rollerblade', 'NN'), ('season', 'NN'), ('today', 'NN'), ('summer', 'NN'), ('rollerblading', 'VBG'), ('belgrade', 'NN'), ('adaciganlija', 'NN')]
[('father', 'RB'), ('day', 'NN')]
[('hype', 'NN'), ('min', 'NN'), ('go', 'VBP'), ('super', 'NN')]
[('eating', 'VBG'), ('breakfast', 'NN'), ('hear', 'JJ'), ('sad', 'JJ'), ('sounds', 'NNS'), ('packing', 'VBG'), ('vacation', 'NN'), ('seemed', 'VBD'), ('shoer', 'JJ'), ('others', 'NNS')]
[('oh', 'UH'), ('noooo', 'JJ'), ('fine', 'JJ'), ('going', 'VBG'), ('flow', 'JJ'), ('epic', 'NN'), ('cantsleptonight', 'NN')]
[('happy', 'JJ'), ('bihday', 'NN'), ('junior', 'JJ'), ('hope', 'NN'), ('best', 'JJS'), ('day', 'NN'), ('bihday', 'NN'), ('cristiano', 'NN')]
[('happy', 'JJ'), ('bihday', 'NN'), ('brithday', 'JJ'), ('day', 'NN'), ('holidays', 'NNS'), ('monday', 'VBP')]
[('freaking', 'VBG'), ('love', 'NN'), ('selfies', 'NNS'), ('alldayeveryday', 'VBP'), ('makes', 'VBZ'), ('goodhairday', 'JJ'), ('makeup', 'NNS'), ('onpoint', 'NN')]
[('find', 'VB'), ('nationalities', 'NNS'), ('happy', 'JJ'), ('days', 'NNS')]

[31] X = df.tweet
    y = df.label

X.head(FIRST_N_RECORDS)
0      oooo tomorrow
1  opened rollerblade season today summer rollerblading belgrade adaciganlija
2                                father day
3                        hype min go super
4      eating breakfast hear sad sounds packing vacation seemed shoer others
5      oh noooo fine going flow epic cantsleptonight
6                        happy bihday junior hope best day bihday cristiano
7      happy bihday brithday day holidays monday
8      freaking love selfies alldayeveryday makes goodhairday makeup onpoint
9                        find nationalities happy days
Name: tweet, dtype: object

```

Model Building and Prediction (with Naïve Bayes)

Tfidf Vectorizer Generation

ICPO3_SentimentAnalysisWithNLTK.ipynb

```

[33] from sklearn.feature_extraction.text import TfidfVectorizer
    vectorizer = TfidfVectorizer()
    feature_vector = vectorizer.fit_transform(X)
    feature_vector.shape

(5972, 12601)

# Vocabulary of the vectorizer
len(vectorizer.vocabulary_)

12601

[35] feature_vector.shape

(5972, 12601)

[36] # idf scores of each word
    idf_scores = dict(zip(vectorizer.get_feature_names(), vectorizer.idf_))

[37] idf_scores.get('love')

3.596930310560657

[38] max(idf_scores.values())

9.001857412172361

```

Top 10 and Bottom 10 words which have the biggest idf and the smallest idf values

```
ICPO3_SentimentAnalysisWithNLTK.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
[39] # The top 10 words which have highest idf
from collections import Counter
counter = Counter(idf_scores)
counter.most_common(10)

[('aa', 9.001857412172361),
 ('aaaaa', 9.001857412172361),
 ('aaaaaand', 9.001857412172361),
 ('aaaaah', 9.001857412172361),
 ('aaahhhhh', 9.001857412172361),
 ('aaaawww', 9.001857412172361),
 ('aah', 9.001857412172361),
 ('aaliyah', 9.001857412172361),
 ('aascf', 9.001857412172361),
 ('aayat', 9.001857412172361)]

# The top 10 words which have lowest idf (These words are so common in the whole corpus)
for key in sorted(counter, key=counter.get, reverse=False)[:FIRST_N_RECORDS]:
    score = idf_scores.get(key)
    print(key, score)

love 3.5969303105660657
day 3.7060431758424426
amp 3.9878943279834305
happy 4.011424825393624
today 4.324366564604643
life 4.3384183180602935
new 4.371994613593898
like 4.39169968467323
one 4.565105877809232
good 4.61983077498478

[41] X_dense = feature_vector.todense()
X_dense.shape
```

Train-Test-Split, Modelling and Prediction

```
ICPO3_SentimentAnalysisWithNLTK.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
[41] X_dense = feature_vector.todense()
X_dense.shape

(5972, 12601)

[42] from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

def summarize_classification(y_test, y_pred):
    acc = accuracy_score(y_test, y_pred, normalize=True)
    num_acc = accuracy_score(y_test, y_pred, normalize=False)
    prec = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')

    print("Length of testing data: ", len(y_test))
    print("accuracy_count : ", num_acc)
    print("accuracy_score : ", acc)
    print("precision_score : ", prec)
    print("recall_score : ", recall)

[44] X_train, X_test, y_train, y_test = train_test_split(X_dense, y, test_size=0.20, random_state=42)

[45] clf = GaussianNB().fit(X_train, y_train)

[46] y_pred = clf.predict(X_test)
```

Modelling Results



The screenshot shows a Jupyter Notebook titled "ICPO3_SentimentAnalysisWithNLTK.ipynb". The interface includes a top bar with file management options (File, Edit, View, Insert, Runtime, Tools, Help) and a status bar indicating "All changes saved". On the right, there are icons for Comment, Share, and a user profile. The notebook has a sidebar with a file explorer and a code editor. The code editor shows the following code:

```
[43]
acc = accuracy_score(y_test, y_pred, normalize=True)
num_acc = accuracy_score(y_test, y_pred, normalize=False)
prec = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

print("Length of testing data: ", len(y_test))
print("accuracy_count : ", num_acc)
print("accuracy_score : ", acc)
print("precision_score : ", prec)
print("recall_score : ", recall)

[44] X_train, X_test, y_train, y_test = train_test_split( X_dense, y, test_size=0.20, random_state=42)

[45] clf = GaussianNB().fit(X_train, y_train)

y_pred = clf.predict(X_test)
summarize_classification(y_test, y_pred)
```

The output of the code is displayed below the code cells:

```
Length of testing data: 1195
accuracy_count : 1081
accuracy_score : 0.9046025104602511
precision_score : 0.9227628408527871
recall_score : 0.9046025104602511
```